AMD Software Tools

May 10, 2006

Agenda

- Distribution Mechanisms
- General Tools Overview
- In Depth View of AMD CodeAnalyst™

Online Tool Distribution

- Public Distribution Web Page
  - **Support** section on amd.com
    - Utilities & Drivers link
    - Software Developers link to AMD Developer Central
  - AMD Developer Central
    - *AMD CodeAnalyst™ Performance Analyzer*
    - *AMD Core Math Library (ACML)*
    - *AMD SimNow™ Simulator*
  - AMD Developer Forums
- Non-Disclosure Agreement (NDA) Web Sites
  - Tools not available to general public
  - Individualized content per NDA number
  - Automatic notification of software updates

Intended Users of AMD Tools

- BIOS / Chipset / OS Development NDA Partners
- Platform Development NDA Partners
- Software Tools Developers - NDA Partners & Public Users
- Software Performance Engineers - NDA Partners & Public
- Soft ware Developers worldwide

BIOS /Chipset / Platform Developers Configuration Validation NDA Tools

- Tools for validating CPU register settings and HT routing configuration implementations for AMD Athlon™ 64 and AMD Opteron™ Processor platforms
- Tools for checking the integrity and accuracy of ACPI on PC platforms that supports ACPI

BIOS /Chipset / Platform Developers NDA Debug Tools

- Hardware debug tool utilizes a special debug operating mode that is powered by the AMD microcode JTAG engine and is used to examine processor registers, memory, cache contents, with additional features to set breakpoints, disassemble code, etc..

BIOS /Chipset / Platform Developers Power Management Validation NDA Tools

- Tools for exercising power management hardware/firmware features
- Tools for validating each hardware/firmware platform supported C-State
- Tools for validating thermal solutions with stress routines to load the system and monitor thermal zones

Software Developer Public Tools

- AMD Power Monitor & AMD Dashboard Demo provide a graphical means for displaying power management activity in a user friendly manner
- AMD CPUInfo was created to provide general users a view into basic system configuration data
- AMD Clock provides a means of monitoring the operating CPU frequency on a per core/node basis
- AMD SimNow™ Simulator provides an accurate model of a computer system from the program, OS, and programmer's point of view
- AMD CodeAnalyst™ Performance Analyzer

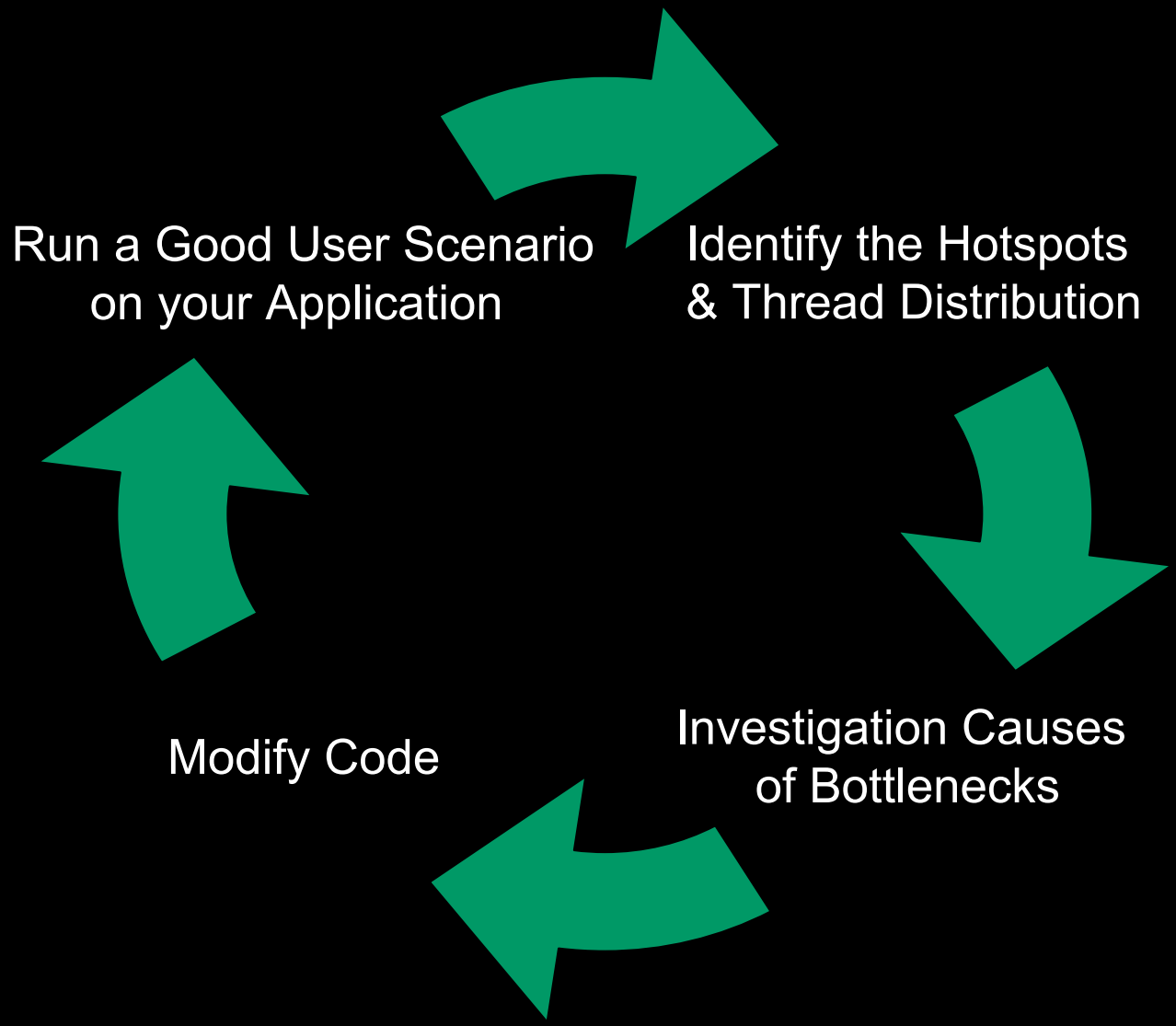
Optimized Numerical Libraries: ACML

- AMD and The Numerical Algorithms Group (NAG) jointly developed the AMD Core Math Library (ACML)
  - *For use with mathematical, engineering, scientific and financial applications as well as general HPC computing*
- ACML is comprised of:
  - *Basic Linear Algebra Subroutines (BLAS) levels 1, 2 and 3*
  - *A wide variety of Fast Fourier Transforms (FFTs)*
  - *Linear Algebra Package (LAPACK)*
- ACML has the following features:
  - *Fortran and C Interfaces*
  - *Highly optimized hand-tuned routines for the AMD64 Instruction Set*
  - *Ability to address single-, double-, single-complex and double-complex data types*
  - *Available for commercially available OSes*
  - *64-bit and 32-bit versions*

What CodeAnalyst™ Provides the Developer

- Overview of the performance Tuning Process using CodeAnalyst™
- CodeAnalyst™ Windows Functionality
- CodeAnalyst™ Linux functionality
- Features under development

The Performance Tuning Process

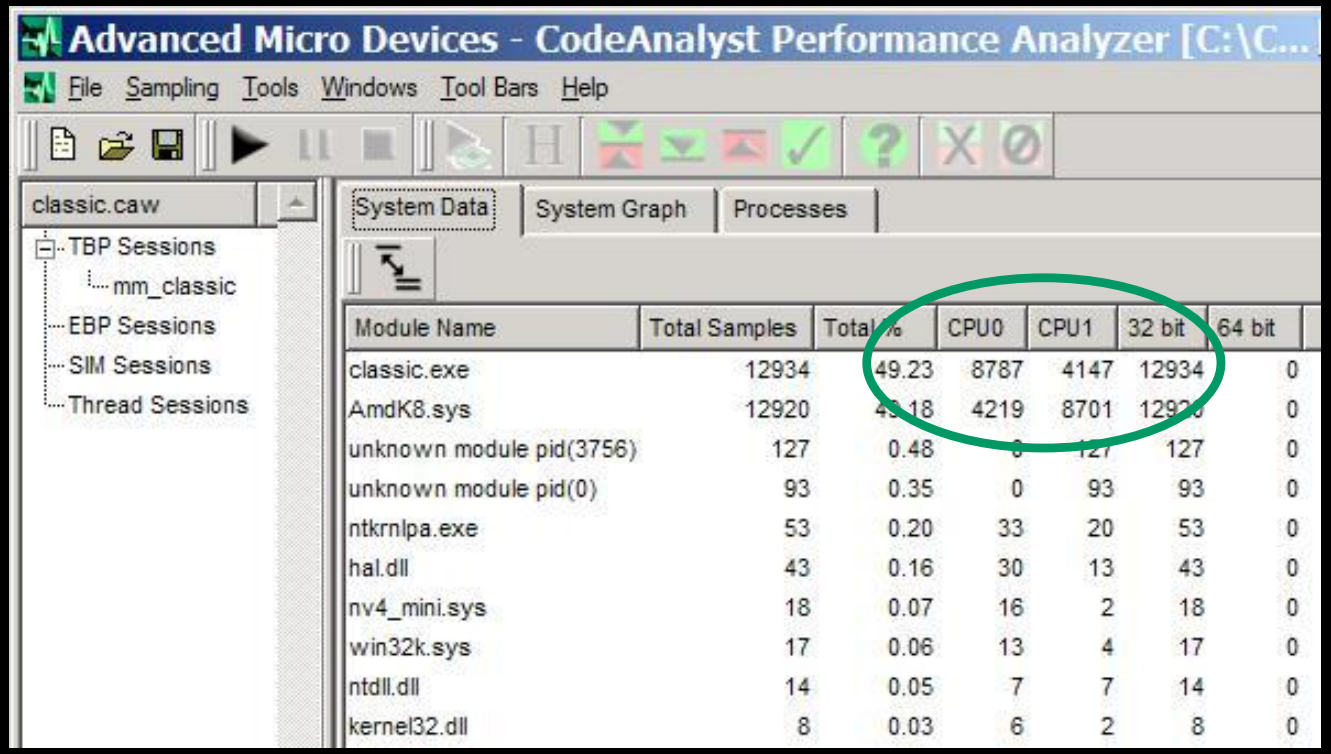


CodeAnalyst™ at a glance

- Timer-based profiling
  - Where is the application or system spending its time?
- Event-based profiling
  - How is the application behaving on the CPU? Memory?
- Thread analysis
  - How are threads behaving on cores? Remote/local access?
- Pipeline simulation
  - Where are delays occurring in the pipeline?

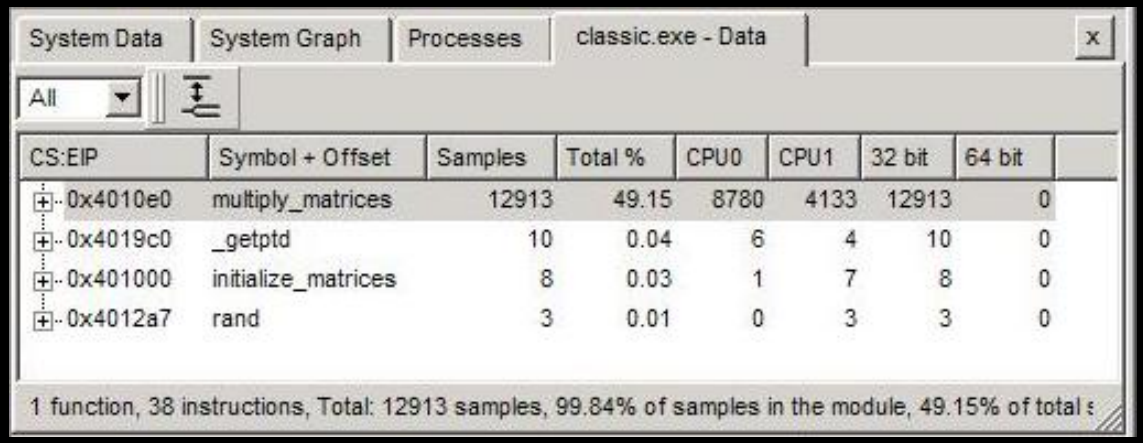
Timer-based profiling

- Display time by module or by process
- Drill down into module or process for more detail
- Shows time on each core



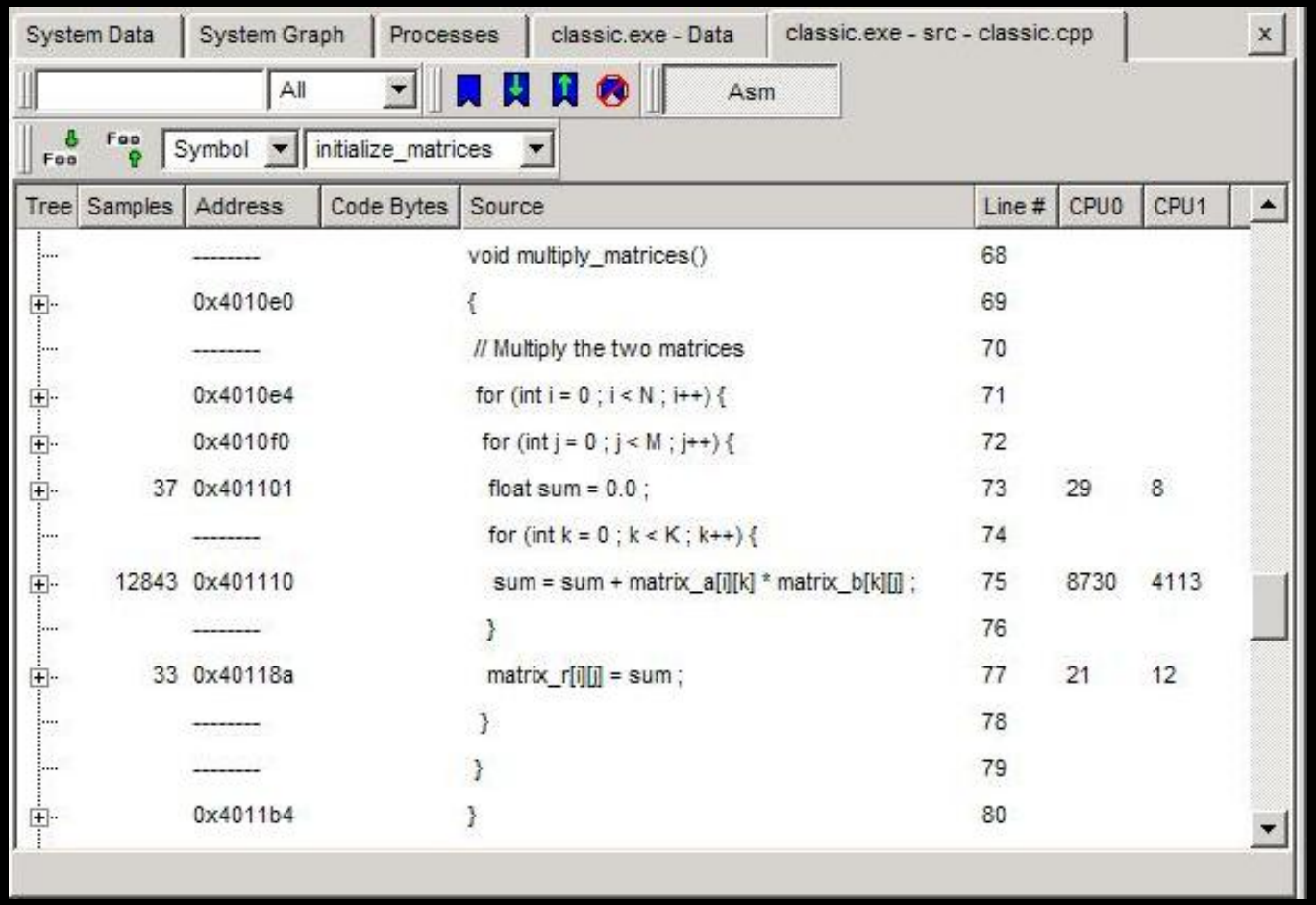
Timer-based profiling

- Drill down to functions within a module
- Display a function-by-function time breakdown
- Identify candidate functions for improvement



Timer-based profiling

- Drill down to source code or even instructions
- Identify code to investigate and tune



Event-based profiling

- Observe how program behaves on CPU and memory
- Develop and test hypothesis about performance issue
- Choose hardware events to measure
- Data display and drill down is similar to Timer-based Profile
- AMD Opteron™ Processors and Athlon™ 64 Processors support profiling 4 simultaneous events
  - For example, one could select
    Unhalted CPU clock cycles (event# 0x76)
    Retired x86 instructions (event# 0xc0)
    Data cache misses (event# 0x41)
    L1 and L2 DTLB misses (event# 0x46)
- AMD Opteron™ and Athlon™ 64 Processors have 78 different performance events
- Provides Timer-Based Profiling thru the 'CPU_CLK_UNHALTED' event

Common used events

- **Data Cache**
- Event 0x40 Data Cache access
- Event 0x41 Data Cache miss
- Event 0x42 Data Cache refill from L2
- Event 0x43 Data Cache refill from system
- Event 0x44 Evicted Line
- Event 0x45 L1 DTLB miss and L2 DTLB hit
- Event 0x46 L1 and L2 DTLB miss
- Event 0x47 Misaligned data reference

Instruction Cache

- Event 0x80 ICache Fetch
- Event 0x81 ICache Miss
- Event 0x84 L1 ITLB miss and L2 ITLB hit
- Event 0x85 L1 ITLB miss and L2 ITLB miss

Branching (FR source)

- 0xC2 Retired Branches including exceptions and interrupts
- 0xC3 Retired branch mispredicted
- 0xC4 Retired taken branches
- 0xC5 Retired taken branches mispredicted

Event combination provides extra info:

- Retired x86 Instruction vs unhalted CPU Clock Cycle will provide IPC.
- Data Cache miss vs Data Cache Access will be Data miss rate.

Penalties

Bottleneck Causes Execution Penalties Notes Annotations	Description	Penalty
Bank conflict	Reference bank conflicted	1+ cycles
Br misprdct	Branch was mispredicted	10+ cycles
BT split wdw	Branch target splits fetch windows	1+ cycles
IC miss	Instruction miss in instruction cache	10+ cycles
C3 misprdct	Jump to 0xc3 opcode caused mispredict	10+ cycles
DC miss	Reference missed in DC	9+ cycle
DTLB miss L0	Reference missed in Data Table L0	2+ cycles
DTLB miss L1	Reference missed in Data Table L1	10+ cycles
DW misalign	Dword reference is misaligned	1+ cycle
ITLB miss L0	Instruction fetch missed in L0 Instruction Table	2+ cycles
ITLB miss L1	Instruction fetch missed in L1 Instruction Table	10+ cycles
Ld/St Q full	Load/Store queue is full	No fixed penalty
No FCOMP	FCOMP does **not** precede FSTSW	Causes code to execute sequentially
No predecode	Instruction was not predecode	7+ cycles
Partial write	Write to a partial register	Causes code to execute sequentially
QW misalign	Qword reference is misaligned	1+ cycle
Serialize inst	Serializing instruction	Huge penalty in terms of cycles. Avoid this penalty.
Stlf 2:0	Store->load forward 11:3 match 2:0 mismatch	Causes code to execute sequentially
Stlf 31:12	Store->load forward 31:12 mismatch	Causes code to execute sequentially
Stlf Ld Misaligned	Store->load forward load is misaligned	Causes code to execute sequentially
Stlf opsize	Store->load forward opsize violation	Causes code to execute sequentially
Stlf St Misaligned	Store->load forward store is misaligned	Causes code to execute sequentially
WC flush	Write combining buffer flushed	No fixed penalty
Wd misalign	Word reference is misaligned	1+ cycle

The Opteron™/Athlon™ 64 Optimization Guide

- Opteron/Athlon64 Optimization Guide is available at http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/25112.PDF
- Audience is both code generator designers/assembly level programmers and C/C+ level programmers
  - emphasis on code generation/assembly level.
- Most optimizations apply to both 64-bit and 32-bit code.
- Note: There is also an overview of the Micro Architecture in Appendix A (includes latencies)

Event-based profiling

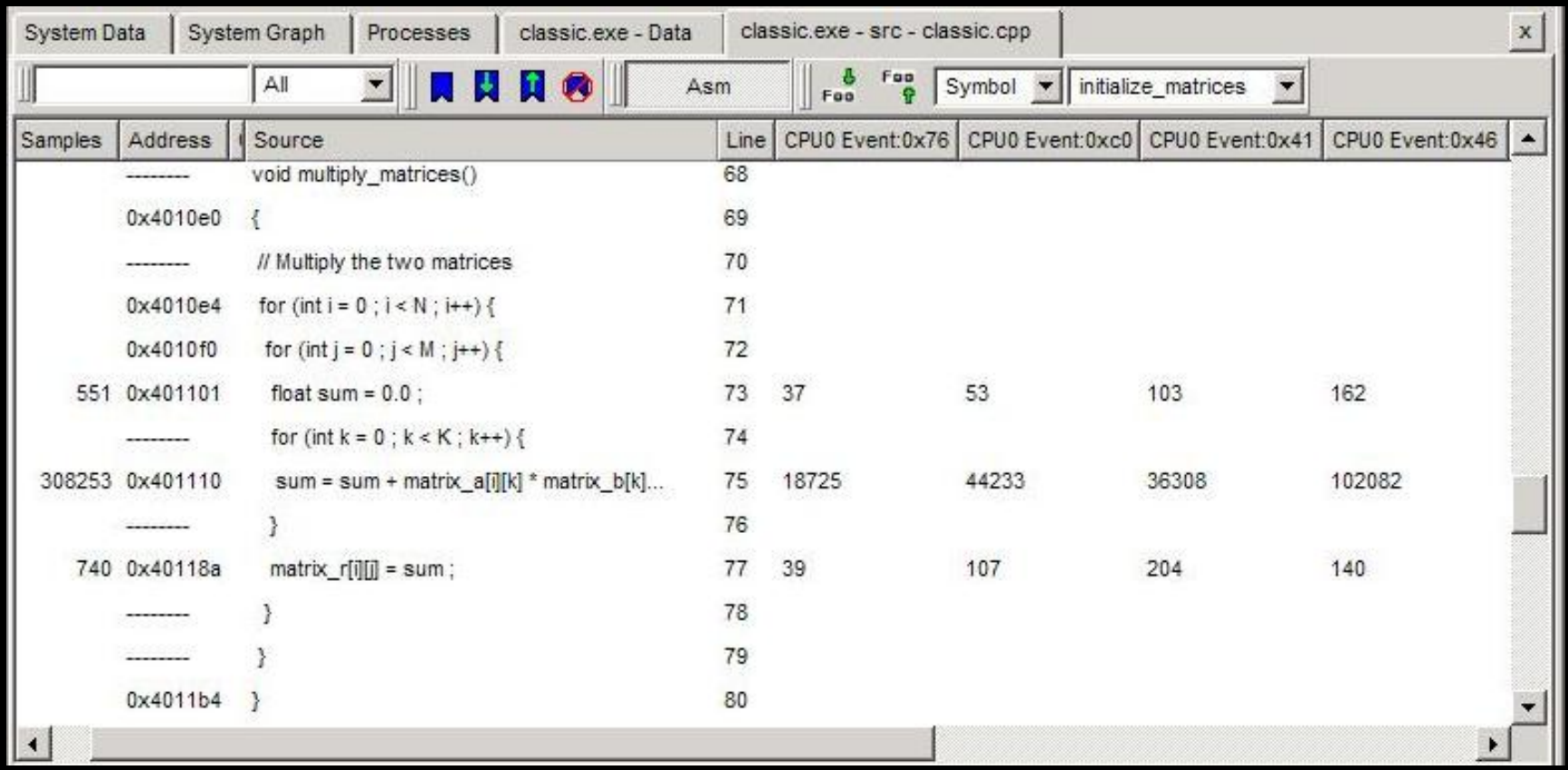
- Data display and drill down is similar to Timer-basedProfile
- Display event breakdown by module, process, core

System Data | System Graph | Processes

Module Name	Total Samples	Total %	CPU0 Event 0x76	CPU0 Event 0xc0	CPU0 Event 0x41	CPU0 Event 0x46	CPU1
classic.exe	309689	98.65	18805	44532	36617	102384	
unknown module pid...	3093	0.99	0	0	0	0	
schedsvc.dll	357	0.11	0	0	0	0	
ntdll.dll	211	0.07	9	77	11	2	
ntkrnlpa.exe	91	0.03	1	84	0	0	
RPCRT4.dll	78	0.02	2	28	5	0	
msvcrt.dll	49	0.02	4	42	0	0	
kernel32.dll	45	0.01	5	15	0	0	

Event-based profiling

- Display event data down to source line or instruction
- High event count and event type helps identify suspect code and issue

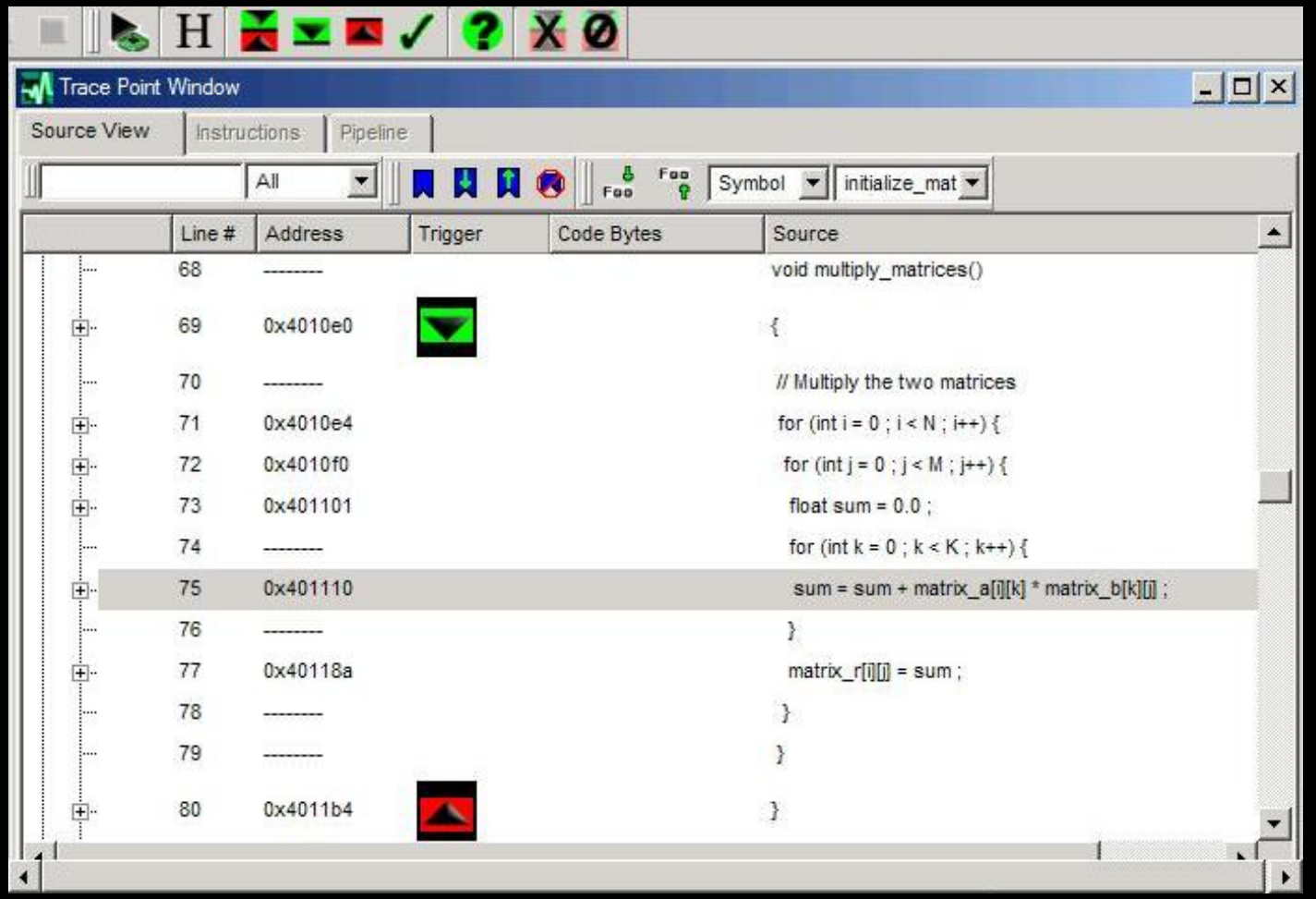


Pipeline simulation

- Investigate issues at very detailed level
- Identify causes for pipeline delay (stalls)
- Find instructions that delay execution
- Uses trace-driven, cycle accurate simulation

Pipeline simulation

- Choose hot spot in code to investigate
- Set start/stop trace points, then click “start” button

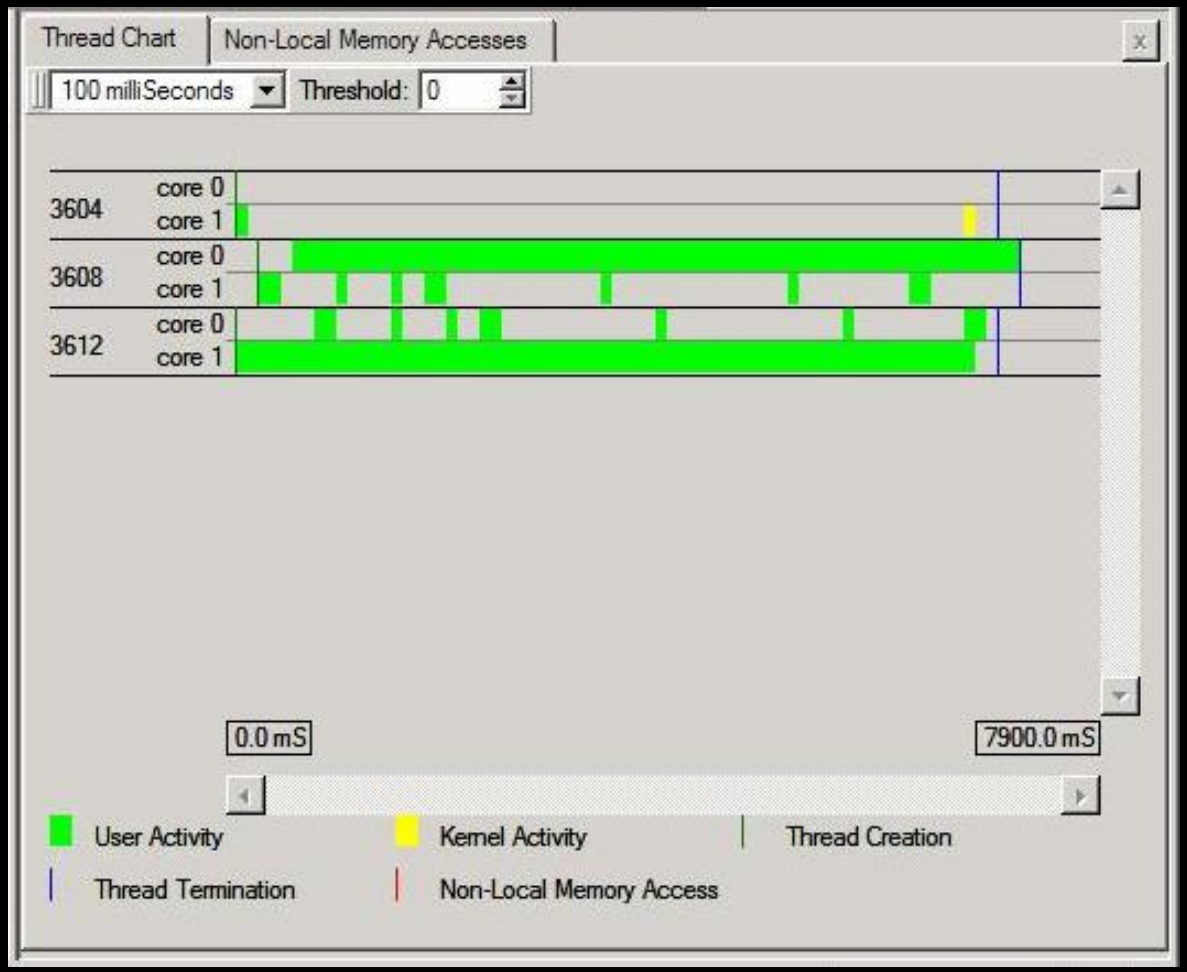


Pipeline simulation

- Display cycle-by-cycle instruction execution history
- Hover mouse over cycle to see cause of delay

Thread analysis

- Display thread activity on cores
- Display remote/local memory access activity



CodeAnalyst™ features on Windows

- Low overhead, system-wide profiling
  - Can profile dynamically loaded modules
  - Can profile kernel mode drivers as well as applications
- Supports multiprocessor profiling
- Command line utilities
  - Supports execution scripts
- Supports profiling of JIT code
  - Java
  - Microsoft .NET
  - Driver JIT code
- Profile control API library
  - Pause and result profiling

CodeAnalyst™ features on Linux

- GUI is consistent with Windows version
  - Display of pipeline simulation results not currently supported
- Uses oprofile for system-wide data collection
  - Can use command line or graphical interface for measurement
  - Can import profile data from oprofile database
- Supports multiprocessor profiling
- Supports profiling of Java applications
- Operates on kernel versions 2.4 and up
- Open source

Present and Planned Investigations by CodeAnalyst Team

- Integration with Microsoft Visual Studio 2005 IDE
- Oprofile Library API
- Call Stack sampling
- Event Multiplexing
- Instruction based sampling
- New Data Display Paradigms for handling large sample data sets in Event Multiplexing and IBS
- Call Graph profile
- Thread profile:
  - Detect thread correctness, dead lock, monitoring synchronization objects etc.

Providing Feedback and Requesting Features

CodeAnalyst™ is available to download at AMD Developer Central at http://developer.amd.com/downloads.aspx

- We welcome feedback, bug reports, requests for help or new features through the Developer Forums on AMD Developer Central